

Attorney's Docket No.: 9269-4

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re: Blaker et al.

Patent No.: 09/852,562

Filed: May 10, 2001

For: CRYPTOGRAPHIC DATA PROCESSING SYSTEMS, COMPUTER PROGRAM PRODUCTS, AND METHODS OF OPERATING SAME IN WHICH A SYSTEM MEMORY IS USED TO TRANSFER INFORMATION BETWEEN A HOST PROCESSOR AND AN ADJUNCT PROCESSOR

Confirmation No.: 6250

Group Art Unit: 2137

Examiner: Paul E. Callahan

Date: March 10, 2006

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Certificate of Mailing

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on March 10, 2006

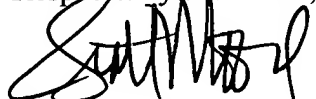

Traci A. Brown

**TRANSMITTAL OF APPEAL BRIEF
(PATENT APPLICATION--37 C.F.R. § 1.192)**

1. Transmitted herewith is the APPEAL BRIEF for the above-identified application, pursuant to the Notice of Appeal received by the US Patent and Trademark Office on **January 9, 2006**.
2. Pursuant to 37 C.F.R. § 1.17(c), the fee for filing the Appeal Brief is:

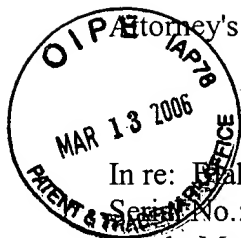
<input type="checkbox"/>	small entity	\$250.00
<input checked="" type="checkbox"/>	other than small entity	\$500.00
3. ☒ Any additional fee or refund may be charged to Deposit Account 50-0220.

Respectfully submitted,


D. Scott Moore
Registration No. 42,011

Myers Bigel Sibley & Sajovec, P.A.
P. O. Box 37428
Raleigh, North Carolina 27627
Telephone: (919) 854-1400
Facsimile: (919) 854-1401
Customer No. 20792

AF
ZMW



Attorney's Docket No.: 9269-4

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re: Baker et al.

Serial No.: 09/852,562

Filed: May 10, 2001

For: CRYPTOGRAPHIC DATA PROCESSING SYSTEMS, COMPUTER PROGRAM PRODUCTS, AND METHODS OF OPERATING SAME IN WHICH A SYSTEM MEMORY IS USED TO TRANSFER INFORMATION BETWEEN A HOST PROCESSOR AND AN ADJUNCT PROCESSOR

Confirmation No.: 6250

Group Art Unit: 2137

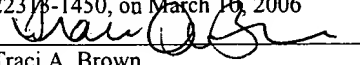
Examiner: Paul E. Callahan

Date: March 10, 2006

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Certificate of Mailing

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on March 10, 2006


Traci A. Brown

APPELLANTS' BRIEF ON APPEAL UNDER 37 C.F.R. §41.37

Sir:

This Appeal Brief is filed pursuant to the "Notice of Appeal to the Board of Patent Appeals and Interferences" mailed January 5, 2006 and received in the U. S. Patent and Trademark Office on January 9, 2006.

Real Party In Interest

The real party in interest is assignee CyberGuard Corporation, Deerfield Beach, Florida.

Related Appeals and Interferences

Appellants are aware of no appeals or interferences that would be affected by the present appeal.

Status of Claims

Appellants appeal the final rejection of Claims 4 - 29, 33 - 45, 49 - 58, 61 - 73, and 77 - 86, which as of the filing date of this Brief remain under consideration. The claims at issue as included in Appellants' response to the Office Action of September 10, 2004 are attached hereto as Appendix A.

Status of Amendments

One substantive response has been filed in the present case: An "Amendment" was filed December 10, 2004 in response to an Office Action mailed September 10, 2004 (hereinafter "First Action"). A final Office Action was mailed October 5, 2005 (hereinafter "Final Action"). Appellants have canceled Claims 1 - 3, 17 - 19, 30 - 32, 46 - 48, 59, 60, and 74 - 76 in prosecuting the present application; therefore, Claims 4 - 29, 33 - 45, 49 - 58, 61 - 73, and 77 - 86 remain for consideration on the present appeal.

Summary of Claimed Subject Matter

Appellants appeal the final rejection of Independent Claims 4, 15, 20, 25, 33, 44, 49, 54, 61, 72, 77, 82.

Independent Claim 4 is directed to a method of operating a cryptographic data processing system that comprises a host processor (FIG. 1, host processor 16), a system memory (FIG. 1, system memory 22) coupled to the host processor (FIG. 1, host processor 16), and a cryptographic processor integrated circuit (FIG. 1, cryptographic processor 14) that comprises a local memory (FIG. 1, memory 36) and is coupled to the host processor (FIG. 1, host processor 16) and the system memory (FIG. 1, system memory 22). The method comprises providing a command queue in the system memory (FIG. 1, command queues 44 and 46; Specification, page 8, lines 25 - 31), loading a command block into the command queue using the host processor (FIG. 2, block 52; Specification, page 10, lines 5 - 7), executing the command block using the cryptographic processor (FIG. 2, block 54; Specification, page 10, lines 10 - 12), and notifying the host processor that the command block has been executed by updating a completion field in the command block using the cryptographic processor (FIG. 2, block 56; Specification, page 10, lines 12 - 14).

Independent Claims 33 and 61 are system and computer program product claims corresponding to method Claim 4. With respect to system Claim 33, the command queues 44 and 46 of FIG. 1 provide structure for the means for providing a command queue in the system memory. The host processor 16 of FIG. 1 provides structure for the means for loading a command block into the command queue. The cryptographic processor 14 of FIG. 1 provides structure for the means for executing the command block. The cryptographic processor 14 of FIG. 1 provides structure for the means for notifying and the means for updating.

Independent Claim 15 is directed to a method of operating a cryptographic data processing system that comprises a host processor (FIG. 1, host processor 16), a system memory (FIG. 1, system memory 22) coupled to the host processor (FIG. 1, host processor 16), and a cryptographic processor integrated circuit (FIG. 1, cryptographic processor 14) that comprises a local memory (FIG. 1, memory 36) and is coupled to the host processor (FIG. 1, host processor 16) and the system memory (FIG. 1, system memory 22). The method comprises providing a command queue in the system memory (FIG. 1, command queues 44 and 46; Specification, page 8, lines 25 - 31), loading a command block into the command queue using the host processor (FIG. 2, block 52; Specification, page 10, lines 5 - 7), setting a value of an interrupt field in the command block to request an interrupt when the command block has been executed (Specification, page 15, lines 9 and 10; FIG. 12A), executing the command block using the cryptographic processor (Specification, page 15, lines 10 - 12; FIG. 11, block 154), and invoking an interrupt using the cryptographic processor after executing the command block if the interrupt field in the command block is set to the value to request the interrupt (Specification, page 15, lines 17 - 20; FIG. 11, block 158).

Independent Claims 44 and 72 are system and computer program product claims corresponding to method Claim 15. With respect to system Claim 44, the command queues 44 and 46 of FIG. 1 provide structure for the means for providing a command queue in the system memory. The host processor 16 of FIG. 1 provides structure for the means for loading a command block into the command queue. The cryptographic processor 14 of FIG. 1 provides structure for the means for setting a value of an interrupt field in the command block. The cryptographic processor 14 of FIG. 1 provides structure for the means for executing the

command block. The cryptographic processor 14 of FIG. 1 provides structure for the means for invoking an interrupt.

Independent Claim 20 is directed to a method of operating a data processing system that comprises a host processor (FIG. 1, host processor 16), a system memory (FIG. 1, system memory 22) coupled to the host processor (FIG. 1, host processor 16), and a cryptographic processor integrated circuit (FIG. 1, cryptographic processor 14) that comprises a local memory (FIG. 1, memory 36) and is coupled to the host processor (FIG. 1, host processor 16) and the system memory (FIG. 1, system memory 22). The method comprises providing a command queue in the system memory (FIG. 1, command queues 44 and 46; Specification, page 8, lines 25 - 31), loading a command block into the command queue using the host processor (FIG. 2, block 52; Specification, page 10, lines 5 - 7), the command block comprising an input data field that contains input data (FIG. 14A), performing an operation based on the input data using the adjunct processor to generate a result (Specification, page 16, lines 20 - 22; FIG. 13, block 164), and storing the result in the input data field such that at least a portion of the input data is overwritten (Specification, page 16, lines 24 - 26; FIG. 13, block 166).

Independent Claims 49 and 77 are system and computer program product claims corresponding to method Claim 20. With respect to system Claim 49, the command queues 44 and 46 of FIG. 1 provide structure for the means for providing a command queue in the system memory. The host processor 16 of FIG. 1 provides structure for the means for loading a command block into the command queue. The cryptographic processor 14 of FIG. 1 provides structure for the means for performing an operation based on the input data. The cryptographic processor 14 of FIG. 1 provides structure for the means for storing the result in the input data field such that at least a portion of the input data is overwritten.

Independent Claim 25 is directed to a method of operating a data processing system that comprises a host processor (FIG. 1, host processor 16), a system memory (FIG. 1, system memory 22) coupled to the host processor (FIG. 1, host processor 16), and a cryptographic processor integrated circuit (FIG. 1, cryptographic processor 14) that comprises a local memory (FIG. 1, memory 36) and is coupled to the host processor (FIG. 1, host processor 16) and the system memory (FIG. 1, system memory 22). The method comprises providing a command queue in the system memory (FIG. 1, command queues 44 and 46; Specification, page 8, lines 25

- 31), providing a read address for the command queue and a write address for the command queue (FIG. 3, read pointer 72, write pointer 74), loading a random number sample into the command queue using the cryptographic processor beginning at the write address (Specification, page 17, lines 28 - 31; FIG. 16, block 172), and reading the random number sample using the host processor beginning at the read address (Specification, page 17, line 31 - page 18, line 2; FIG. 16, block 174).

Independent Claims 54 and 82 are system and computer program product claims corresponding to method Claim 25. With respect to system Claim 54, the command queues 44 and 46 of FIG. 1 provide structure for the means for providing a command queue in the system memory. The read pointer 72 and write pointer 74 of FIG. 3 provide structure for the means for providing a read address for the command queue and a write address for the command queue. The cryptographic processor 14 of FIG. 1 provides structure for the means for loading a random number sample into the command queue. The host processor 16 provides structure for the means for reading the random number sample.

Grounds of Rejection to be Reviewed on Appeal

Independent Claims 4, 33, and 61 stand rejected under 35 U.S.C. §103(a) as being unpatentable over European Patent Application EP 0 945 788 A2 of Hocevar et al. (hereinafter "Hocevar") in view of U. S. Patent No. 5,706,489 to Chi et al. (hereinafter "Chi").

Independent Claims 15, 44, and 72 stand rejected under 35 U.S.C. §102(b) as being anticipated by U. S. Patent No. 6,075,456 to Hussain et al. (hereinafter "Hussain").

Independent Claims 20, 25, 49, 54, 77, and 82 stand rejected under 35 U.S.C. §102(b) as being anticipated by Hocevar.

Argument

I. Introduction to 35 U.S.C. §102/§103 Analysis

Under 35 U.S.C. § 102, "a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." M.P.E.P. § 2131 (quoting *Verdegaal Bros. v. Union Oil Co.*, 814 F.2d 628, 631, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987)). "Anticipation under 35 U.S.C. § 102 requires the disclosure in a

single piece of prior art of each and every limitation of a claimed invention." *Apple Computer Inc. v. Articulate Sys. Inc.*, 57 U.S.P.Q.2d 1057, 1061 (Fed. Cir. 2000). "The fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic. To establish inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.'" M.P.E.P. § 2112 (citations omitted).

A finding of anticipation further requires that there must be no difference between the claimed invention and the disclosure of the cited reference as viewed by one of ordinary skill in the art. See *Scripps Clinic & Research Foundation v. Genentech Inc.*, 927 F.2d 1565, 1576, 18 U.S.P.Q.2d 1001, 1010 (Fed. Cir. 1991). In particular, the Court of Appeals for the Federal Circuit held that a finding of anticipation requires absolute identity for each and every element set forth in the claimed invention. See *Trintec Indus. Inc. v. Top-U.S.A. Corp.*, 63 U.S.P.Q.2d 1597 (Fed. Cir. 2002). Additionally, the cited prior art reference must be enabling, thereby placing the allegedly disclosed matter in the possession of the public. *In re Brown*, 329 F.2d 1006, 1011, 141 U.S.P.Q. 245, 249 (C.C.P.A. 1964). Thus, the prior art reference must adequately describe the claimed invention so that a person of ordinary skill in the art could make and use the invention.

A determination under §103 that an invention would have been obvious to someone of ordinary skill in the art is a conclusion of law based on fact. *Panduit Corp. v. Dennison Mfg. Co.* 810 F.2d 1593, 1 U.S.P.Q.2d 1593 (Fed. Cir. 1987), *cert. denied*, 107 S.Ct. 2187. After the involved facts are determined, the decision maker must then make the legal determination of whether the claimed invention as a whole would have been obvious to a person having ordinary skill in the art at the time the invention was unknown, and just before it was made. *Id.* at 1596. The United States Patent and Trademark Office (USPTO) has the initial burden under §103 to establish a *prima facie* case of obviousness. *In re Fine*, 837 F.2d 1071, 5 U.S.P.Q.2d 1596, 1598 (Fed. Cir. 1988).

To establish a *prima facie* case of obviousness, the prior art reference or references when combined must teach or suggest *all* the recitations of the claims, and there must be some

suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. M.P.E.P. §2143. The mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination. M.P.E.P. §2143.01, citing *In re Mills*, 916 F.2d 680, 16 U.S.P.Q.2d 1430 (Fed. Cir. 1990). As emphasized by the Court of Appeals for the Federal Circuit, to support combining references, evidence of a suggestion, teaching, or motivation to combine must be **clear and particular**, and this requirement for clear and particular evidence is not met by broad and conclusory statements about the teachings of references. *In re Dembiczak*, 50 U.S.P.Q.2d 1614, 1617 (Fed. Cir. 1999). In another decision, the Court of Appeals for the Federal Circuit has stated that, to support combining or modifying references, there must be **particular** evidence from the prior art as to the reason the skilled artisan, with no knowledge of the claimed invention, would have selected these components for combination in the manner claimed. *In re Kotzab*, 55 U.S.P.Q.2d 1313, 1317 (Fed. Cir. 2000).

Appellants respectfully submit that the pending independent claims are patentable over the cited reference(s) for at least the reason that the cited reference(s) do not disclose or suggest each of the recitations of the independent claims. The patentability of the pending claims is discussed in detail hereinafter.

A. Independent Claims 4, 33, and 61 are Patentable over Hocevar and Chi

Independent Claim 4 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Hocevar in view of Chi. Independent Claim 4 recites, in part:

- providing a command queue in the system memory;
- loading a command block into the command queue using the host processor;
- executing the command block using the cryptographic processor; and
- notifying the host processor that the command block has been executed by updating a completion field in the command block using the cryptographic processor.

Claims 33 and 61 include similar recitations. The First Action acknowledges that Hocevar does not disclose or suggest notifying the host processor that the command block has been executed

by updating a completion field in the command block using the cryptographic processor, but alleges that Chi provides the missing teachings at col. 7, line 59 through col. 8, line 5 and col. 8, lines 42 - 50. (First Action, pages 11 - 12).

Appellants respectfully submit that the first passage of Chi cited in the First Action (col. 7, line 59 through col. 8, line 5) describes a halt command for a parallel processor and appears to include no teaching or suggestion related to notifying a host processor that a command block has been executed by updating a completion field as recited in Claim 4, as amended. The second passage of Chi cited in the Office Action (col. 8, lines 42 - 50) describes setting a field in a header block 215 to indicate when an asynchronous operation is completed. The header block 215 is described at col. 4, lines 37 - 47 and appears not to be used to communicate information from the parallel instruction execution (PIE) facility 120 to the processing unit 110, but is instead used to communicate information from the processing unit 110, to the PIE facility 120. The Final Action continues to maintain that updating the PHB header block 215 corresponds to the recitation of notifying the host processor that a command block has been executed. (Final Action, page 2). Appellants respectfully disagree, however, because the PHB header block 215 is not used to communicate information from the PIE facility 120 to the processing unit 110, but is instead to used to communicate information from the processing unit 110 to the PIE facility as described at col. 4, lines 37 - 47. The Final Action does not address this argument or this passage from Chi.

Accordingly, for at least the foregoing reasons, Appellants respectfully submit that independent Claims 4, 33, and 61 are patentable over Hocevar in view of Chi, and that Claims 5 - 10, 12 - 14, 34 - 39, 41 - 43, 62 - 67, and 69 - 71 are patentable at least per the patentability of independent Claims 4, 33, and 61. Appellants respectfully request that the rejection of independent Claims 4, 33, and 61 be reversed based on the failure of the Examiner to establish a *prima facie* case of obviousness under 35 U.S.C. §103 for at least these reasons.

B. Independent Claims 15, 44, and 72 are Patentable over Hussain

Independent Claims 15, 44, and 72 stand rejected under 35 U.S.C. §102(b) as being anticipated by Hussain. Independent Claim 15 recites, in part:

providing a command queue in the system memory;

- loading a command block into the command queue using the host processor;
- setting a value of an interrupt field in the command block to request an interrupt when the command block has been executed;
- executing the command block using the cryptographic processor; and
- invoking an interrupt using the cryptographic processor after executing the command block if the interrupt field in the command block is set to the value to request the interrupt.

Claims 44 and 72 include similar recitations. The First Action cites col. 8, lines 53 - 59 of Hussain as disclosing invoking an interrupt using the cryptographic processor after executing the command block if the interrupt field in the command block is set to the value to request the interrupt. (First Action, page 10). Appellants respectfully disagree with this interpretation of Hussain's teachings. The aforementioned passage of Hussain cited in the First Action pertains to the rendering engine 104 generating an interrupt for the host processor rather than the graphics processor 114. The rendering engine 104 is used for address translation (Hussain, col. 4, lines 2 - 8) and, therefore, is unrelated to a cryptographic processor as recited in Claim 15. The Final Action continues to maintain the rejection based on the operation of the rendering engine 104 (Final Action, page 3), but Appellants continue to maintain that the rendering engine 104 of Hussain is not a cryptographic processor as recited in Claim 15.

Accordingly, for at least the foregoing reasons, Appellants respectfully submit that independent Claims 15, 44, and 72 are patentable over Hussain, and that Claims 16, 45, and 73 are patentable at least per the patentability of independent Claims 15, 44, and 72. Appellants respectfully request that the rejection of independent Claims 15, 44, and 72 be reversed based on the failure of the Examiner to establish a *prima facie* case of anticipated under 35 U.S.C. §102 for at least these reasons.

C. Independent Claims 20, 49, and 77 are Patentable over Hocevar

Independent Claim 20 stands rejected under 35 U.S.C. §102(b) as being anticipated by Hocevar. Independent Claim 20 recites, in part:

- providing a command queue in the system memory;
- loading a command block into the command queue using the host processor, the command block comprising an input data field that contains input data;

performing an operation based on the input data using the adjunct processor to generate a result; and
storing the result in the input data field such that at least a portion of the input data is overwritten.

Claims 49 and 77 include similar recitations. The First Action asserts that independent Claim 20 is rejected for the same reasons as independent Claim 4. (First Action, page 7). Appellants note, however, that Claim 20 includes the highlighted recitation "storing the result in the input data field such that at least a portion of the input data is overwritten." Appellants respectfully submit that Hocevar contains no disclosure or suggestion of at least this highlighted recitation.

Accordingly, for at least the foregoing reasons, Appellants respectfully submit that independent Claims 20, 49, and 77 are patentable over Hocevar, and that Claims 21 - 24, 50 - 53, and 78 - 81 are patentable at least per the patentability of independent Claims 20, 49, and 77. Appellants respectfully request that the rejection of independent Claims 20, 49, and 77 be reversed based on the failure of the Examiner to establish a *prima facie* case of anticipation under 35 U.S.C. §102 for at least these reasons.

D. Independent Claims 25, 54, and 82 are Patentable over Hocevar

Independent Claim 25 stands rejected under 35 U.S.C. §102(b) as being anticipated by Hocevar. Independent Claim 25 recites, in part:

providing a command queue in the system memory;
providing a read address for the command queue and a write address for the command queue;
loading a random number sample into the command queue using the cryptographic processor beginning at the write address; and
reading the random number sample using the host processor beginning at the read address.

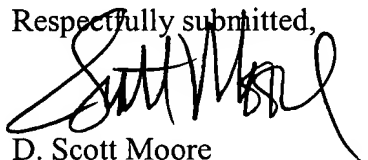
Claims 54 and 82 include similar recitations. The First Action cites col. 3, line 38 through col. 5, line 21 of Hocevar as disclosing the recitations of Claim 25. Appellants respectfully disagree with this interpretation of Hocevar's teachings as Appellants can find no mention of, at least, loading a random number sample into the command queue and reading the random number sample recitations. In particular, Appellants cannot find any reference to a random number sample in Hocevar's disclosure.

Accordingly, for at least the foregoing reasons, Appellants respectfully submit that independent Claims 25, 54, and 82 are patentable over Hovevar, and that Claims 26 - 29, 55 - 58, and 83 - 86 are patentable at least per the patentability of independent Claims 25, 54, and 82. Appellants respectfully request that the rejection of independent Claims 25, 54, and 82 be reversed based on the failure of the Examiner to establish a *prima facie* case of anticipation under 35 U.S.C. §102 for at least these reasons.

II. Conclusion

In summary, Appellants respectfully submit that, with respect to Claims 4, 15, 20, 25, 33, 44, 49, 54, 61, 72, 77, 82, the cited reference(s) does not teach all of the recitations of the claims. Accordingly, Appellants respectfully request reversal of the rejection of Claims 4, 15, 20, 25, 33, 44, 49, 54, 61, 72, 77, 82 based on the cited reference(s) and the claims that depend therefrom.

Respectfully submitted,



D. Scott Moore
Registration No. 42,011

Myers Bigel Sibley & Sajovec, P.A.
P. O. Box 37428
Raleigh, North Carolina 27627
Telephone: (919) 854-1400
Facsimile: (919) 854-1401
Customer No. 20792

APPENDIX A

1 - 3. (Canceled)

4. (Previously presented) A method of operating a cryptographic data processing system that comprises a host processor, a system memory coupled to the host processor, and a cryptographic processor integrated circuit that comprises a local memory and is coupled to the host processor and the system memory, the method comprising:

- providing a command queue in the system memory;
- loading a command block into the command queue using the host processor;
- executing the command block using the cryptographic processor; and
- notifying the host processor that the command block has been executed by updating a completion field in the command block using the cryptographic processor.

5. (Original) A method as recited in Claim 4, further comprising:

- providing a read address for the command queue and a write address for the command queue;

- wherein loading the command block into the command queue using the host processor comprises loading the command block into the command queue using the host processor beginning at the write address, and wherein executing the command block using the cryptographic processor comprises executing the command block using the cryptographic processor beginning at the read address.

6. (Original) A method as recited in Claim 5, wherein loading the command block into the command queue using the host processor beginning at the write address comprises:

- determining if the write address plus an amount corresponding to a size of a single command block equals the read address; and

- loading the command block into the command queue using the host processor beginning at the write address if the write address plus the amount corresponding to the size of the single command block does not equal the read address.

7. (Original) A method as recited in Claim 6, further comprising:
incrementing the write address by the amount corresponding to the size of a single command block using the host processor after loading the command block into the command queue using the host processor beginning at the write address if the write address plus the amount corresponding to the size of the single command block does not equal the read address.
8. (Original) A method as recited in Claim 5, wherein executing the command block using the cryptographic processor beginning at the read address comprises:
determining whether the read address is equal to the write address; and
executing the command block using the cryptographic processor beginning at the read address if the read address is not equal to the write address.
9. (Original) A method as recited in Claim 8, further comprising:
incrementing the read address by an amount corresponding to a size of a single command block using the cryptographic processor after executing the command block using the cryptographic processor beginning at the read address.
10. (Original) A method as recited in Claim 4, wherein notifying the host processor that the command block has been executed comprises invoking an interrupt using the cryptographic processor after executing the command block.
11. (Canceled)
12. (Previously presented) A method as recited in Claim 4, further comprising:
providing a periodic interrupt; and
reading the completion field using the host processor upon invocation of the periodic interrupt.

13. (Original) A method as recited in Claim 4, wherein notifying the host processor that the command block has been executed comprises:

setting a timer after loading the command block into the command queue using the host processor; and

checking whether the command block has been executed after expiration of the timer.

14. (Original) A method as recited in Claim 4, further comprising:
loading at least one operand from the command queue to the local memory;
performing at least one operation on the at least one operand to generate a result in the local memory; and

storing the result generated in the local memory in the command queue.

15. (Original) A method of operating a cryptographic data processing system that comprises a host processor, a system memory coupled to the host processor, and a cryptographic processor integrated circuit that is coupled to the host processor and the system memory, the method comprising:
providing a command queue in the system memory;
loading a command block into the command queue using the host processor;
setting a value of an interrupt field in the command block to request an interrupt when the command block has been executed;

executing the command block using the cryptographic processor; and

invoking an interrupt using the cryptographic processor after executing the command block if the interrupt field in the command block is set to the value to request the interrupt.

16. (Original) A method as recited in Claim 15, further comprising:
storing error information in the command block that is associated with executing the command block using the cryptographic processor.

17 - 19. (Canceled)

20. (Original) A method of operating a data processing system that comprises a host processor, a system memory coupled to the host processor, and an adjunct processor integrated circuit that is coupled to the host processor and the system memory, the method comprising:

- providing a command queue in the system memory;
- loading a command block into the command queue using the host processor, the command block comprising an input data field that contains input data;
- performing an operation based on the input data using the adjunct processor to generate a result; and
- storing the result in the input data field such that at least a portion of the input data is overwritten.

21. (Original) A method as recited in Claim 20, wherein the data processing system comprises a cryptographic data processing system, the adjunct processor integrated circuit comprises a cryptographic processor integrated circuit, and performing the operation based on the input data comprises:

- performing a hash operation based on the input data using the cryptographic processor to generate a hash value.

22. (Original) A method as recited in Claim 21, wherein storing the result in the input data field comprises:

- storing the hash value in the input data field such that the at least a portion of the input data is overwritten.

23. (Original) A method as recited in Claim 21, wherein the command block further comprises an input pointer field that contains an address in the system memory of an incoming packet and wherein performing the hash operation comprises:

- performing the hash operation based on the input data and the incoming packet using the cryptographic processor to generate the hash value.

24. (Original) A method as recited in Claim 23, wherein the command block further comprises an output pointer field that contains an address in the system memory for storing a decrypted packet, the method further comprising:

- decrypting the incoming packet using the cryptographic processor to generate the decrypted packet;

- attaching the hash value to the decrypted packet; and

- storing the decrypted packet with the attached hash value at the address in the system memory contained in the output pointer field.

25. (Original) A method of operating a cryptographic data processing system that comprises a host processor, a system memory coupled to the host processor, and a cryptographic processor integrated circuit that comprises a local memory and is coupled to the host processor and the system memory, the method comprising:

- providing a command queue in the system memory;

- providing a read address for the command queue and a write address for the command queue;

- loading a random number sample into the command queue using the cryptographic processor beginning at the write address; and

- reading the random number sample using the host processor beginning at the read address.

26. (Original) A method as recited in Claim 25, wherein loading the random number sample into the command queue using the cryptographic processor beginning at the write address comprises:

- determining if the write address plus an amount corresponding to a size of a single random number sample equals the read address; and

- loading the random number sample into the command queue using the cryptographic processor beginning at the write address if the write address plus the amount corresponding to the size of the single random number sample does not equal the read address.

27. (Original) A method as recited in Claim 26, further comprising:
incrementing the write address by the amount corresponding to the size of a single random number sample using the cryptographic processor after loading the random number sample into the command queue using the cryptographic processor beginning at the write address if the write address plus the amount corresponding to the size of the single random number sample does not equal the read address.

28. (Original) A method as recited in Claim 25, wherein reading the random number sample using the host processor beginning at the read address comprises:
determining whether the read address is equal to the write address; and
reading the random number sample using the host processor beginning at the read address if the read address is not equal to the write address.

29. (Original) A method as recited in Claim 28, further comprising:
incrementing the read address by an amount corresponding to a size of a single random number sample using the host processor after reading the random number sample using the host processor beginning at the read address.

30 - 32. (Canceled)

33. (Previously presented) A cryptographic data processing system that comprises a host processor, a system memory coupled to the host processor, and a cryptographic processor integrated circuit that comprises a local memory and is coupled to the host processor and the system memory, the system further comprising:

means for providing a command queue in the system memory;
means for loading a command block into the command queue using the host processor;
means for executing the command block using the cryptographic processor; and
means for notifying the host processor that the command block has been executed, the means for notifying comprising means for updating a completion field in the command block using the cryptographic processor.

34. (Original) A cryptographic data processing system as recited in Claim 33, further comprising:

means for providing a read address for the command queue and a write address for the command queue;

wherein the means for loading the command block into the command queue using the host processor comprises means for loading the command block into the command queue using the host processor beginning at the write address, and wherein the means for executing the command block using the cryptographic processor comprises means for executing the command block using the cryptographic processor beginning at the read address.

35. (Original) A cryptographic data processing system as recited in Claim 34, wherein the means for loading the command block into the command queue using the host processor beginning at the write address comprises:

means for determining if the write address plus an amount corresponding to a size of a single command block equals the read address; and

means for loading the command block into the command queue using the host processor beginning at the write address if the write address plus the amount corresponding to the size of the single command block does not equal the read address.

36. (Original) A cryptographic data processing system as recited in Claim 35, further comprising:

means for incrementing the write address by the amount corresponding to the size of a single command block using the host processor if the write address plus the amount corresponding to the size of the single command block does not equal the read address, the means for incrementing being responsive to the means for loading the command block into the command queue using the host processor beginning at the write address.

37. (Original) A cryptographic data processing system as recited in Claim 34, wherein the means for executing the command block using the cryptographic processor beginning at the read address comprises:

means for determining whether the read address is equal to the write address; and

means for executing the command block using the cryptographic processor beginning at the read address if the read address is not equal to the write address.

38. (Original) A cryptographic data processing system as recited in Claim 37, further comprising:

means for incrementing the read address by an amount corresponding to a size of a single command block using the cryptographic processor, the means for incrementing being responsive to the means for executing the command block using the cryptographic processor beginning at the read address.

39. (Original) A cryptographic data processing system as recited in Claim 33, wherein the means for notifying the host processor that the command block has been executed comprises means for invoking an interrupt using the cryptographic processor after executing the command block.

40. (Canceled)

41. (Currently amended) A cryptographic data processing system as recited in Claim 33, further comprising:

means for providing a periodic interrupt; and

means for reading the completion field using the host processor upon invocation of the periodic interrupt.

42. (Original) A cryptographic data processing system as recited in Claim 33, wherein the means for notifying the host processor that the command block has been executed comprises:

means for setting a timer after loading the command block into the command queue using the host processor; and

means for checking whether the command block has been executed after expiration of the timer.

43. (Original) A cryptographic data processing system as recited in Claim 33, further comprising:

means for loading at least one operand from the command queue to the local memory;

means for performing at least one operation on the at least one operand to generate a result in the local memory; and

means for storing the result generated in the local memory in the command queue.

44. (Original) A cryptographic data processing system that comprises a host processor, a system memory coupled to the host processor, and a cryptographic processor integrated circuit that is coupled to the host processor and the system memory, the system further comprising:

means for providing a command queue in the system memory;

means for loading a command block into the command queue using the host processor;

means for setting a value of an interrupt field in the command block to request an interrupt when the command block has been executed;

means for executing the command block using the cryptographic processor; and

means for invoking an interrupt using the cryptographic processor after executing the command block if the interrupt field in the command block is set to the value to request the interrupt.

45. (Original) A cryptographic data processing system as recited in Claim 44, further comprising:

means for storing error information in the command block that is associated with executing the command block using the cryptographic processor.

46 - 48. (Canceled)

49. (Original) A data processing system that comprises a host processor, a system memory coupled to the host processor, and an adjunct processor integrated circuit that is coupled to the host processor and the system memory, the system further comprising:

means for providing a command queue in the system memory;

means for loading a command block into the command queue using the host processor, the command block comprising an input data field that contains input data;

means for performing an operation based on the input data using the adjunct processor to generate a result; and

means for storing the result in the input data field such that at least a portion of the input data is overwritten.

50. (Original) A data processing system as recited in Claim 49, wherein the data processing system comprises a cryptographic data processing system, the adjunct processor integrated circuit comprises a cryptographic processor integrated circuit, and the means for performing the operation based on the input data comprises:

means for performing a hash operation based on the input data using the cryptographic processor to generate a hash value.

51. (Original) A data processing system as recited in Claim 50, wherein the means for storing the result in the input data field comprises:

means for storing the hash value in the input data field such that the at least a portion of the input data is overwritten.

52. (Original) A data processing system as recited in Claim 50, wherein the command block further comprises an input pointer field that contains an address in the system memory of an incoming packet and wherein the means for performing the hash operation comprises:

means for performing the hash operation based on the input data and the incoming packet using the cryptographic processor to generate the hash value.

53. (Original) A data processing system as recited in Claim 52, wherein the command block further comprises an output pointer field that contains an address in the system memory for storing a decrypted packet, the data processing system further comprising:

means for decrypting the incoming packet using the cryptographic processor to generate the decrypted packet;

means for attaching the hash value to the decrypted packet; and

means for storing the decrypted packet with the attached hash value at the address in the system memory contained in the output pointer field.

54. (Original) A cryptographic data processing system that comprises a host processor, a system memory coupled to the host processor, and a cryptographic processor integrated circuit that comprises a local memory and is coupled to the host processor and the system memory, the system further comprising:

means for providing a command queue in the system memory;

means for providing a read address for the command queue and a write address for the command queue;

means for loading a random number sample into the command queue using the cryptographic processor beginning at the write address; and

means for reading the random number sample using the host processor beginning at the read address.

55. (Original) A cryptographic data processing system as recited in Claim 54, wherein the means for loading the random number sample into the command queue using the cryptographic processor beginning at the write address comprises:

means for determining if the write address plus an amount corresponding to a size of a single random number sample equals the read address; and

means for loading the random number sample into the command queue using the cryptographic processor beginning at the write address if the write address plus the amount corresponding to the size of the single random number sample does not equal the read address.

56. (Original) A cryptographic data processing system as recited in Claim 55, further comprising:

means for incrementing the write address by the amount corresponding to the size of a single random number sample using the cryptographic processor if the write address plus the amount corresponding to the size of the single random number sample does not equal the read address, the means for incrementing being responsive to the means for loading the random number sample into the command queue using the cryptographic processor beginning at the write address.

57. (Original) A cryptographic data processing system as recited in Claim 54, wherein the means for reading the random number sample using the host processor beginning at the read address comprises:

means for determining whether the read address is equal to the write address; and

means for reading the random number sample using the host processor beginning at the read address if the read address is not equal to the write address.

58. (Original) A cryptographic data processing system as recited in Claim 57, further comprising:

means for incrementing the read address by an amount corresponding to a size of a single random number sample using the host processor, the means for incrementing being responsive to the means for reading the random number sample using the host processor beginning at the read address.

59 - 60. (Canceled)

61. (Previously presented) A computer program product for operating a cryptographic data processing system that comprises a host processor, a system memory coupled to the host processor, and a cryptographic processor integrated circuit that comprises a local memory and is

coupled to the host processor and the system memory, the computer program product comprising:

a computer readable program medium having computer readable program code embodied therein, the computer readable program code comprising:

computer readable program code for providing a command queue in the system memory;
computer readable program code for loading a command block into the command queue using the host processor;

computer readable program code for executing the command block using the cryptographic processor; and

computer readable program code for notifying the host processor that the command block has been executed, the computer readable program code for notifying comprising computer readable program code for updating a completion field in the command block using the cryptographic processor.

62. (Original) A computer program product as recited in Claim 61, further comprising:

computer readable program code for providing a read address for the command queue and a write address for the command queue;

wherein the computer readable program code for loading the command block into the command queue using the host processor comprises computer readable program code for loading the command block into the command queue using the host processor beginning at the write address, and wherein the computer readable program code for executing the command block using the cryptographic processor comprises computer readable program code for executing the command block using the cryptographic processor beginning at the read address.

63. (Original) A computer program product as recited in Claim 62, wherein the computer readable program code for loading the command block into the command queue using the host processor beginning at the write address comprises:

computer readable program code for determining if the write address plus an amount corresponding to a size of a single command block equals the read address; and

computer readable program code for loading the command block into the command queue using the host processor beginning at the write address if the write address plus the amount corresponding to the size of the single command block does not equal the read address.

64. (Original) A computer program product as recited in Claim 63, further comprising:

computer readable program code for incrementing the write address by the amount corresponding to the size of a single command block using the host processor if the write address plus the amount corresponding to the size of the single command block does not equal the read address, the computer readable program code for incrementing being responsive to the computer readable program code for loading the command block into the command queue using the host processor beginning at the write address.

65. (Original) A computer program product as recited in Claim 62, wherein the computer readable program code for executing the command block using the cryptographic processor beginning at the read address comprises:

computer readable program code for determining whether the read address is equal to the write address; and

computer readable program code for executing the command block using the cryptographic processor beginning at the read address if the read address is not equal to the write address.

66. (Original) A computer program product as recited in Claim 65, further comprising:

computer readable program code for incrementing the read address by an amount corresponding to a size of a single command block using the cryptographic processor, the computer readable program code for incrementing being responsive to the computer readable program code for executing the command block using the cryptographic processor beginning at the read address.

67. (Original) A computer program product as recited in Claim 61, wherein the computer readable program code for notifying the host processor that the command block has been executed comprises computer readable program code for invoking an interrupt using the cryptographic processor after executing the command block.

68. (Canceled)

69. (Previously presented) A computer program product as recited in Claim 61, further comprising:

computer readable program code for providing a periodic interrupt; and

computer readable program code for reading the completion field using the host processor upon invocation of the periodic interrupt.

70. (Original) A method as recited in Claim 61, wherein the computer readable program code for notifying the host processor that the command block has been executed comprises:

computer readable program code for setting a timer after loading the command block into the command queue using the host processor; and

computer readable program code for checking whether the command block has been executed after expiration of the timer.

71. (Original) A computer program product as recited in Claim 61, further comprising:

computer readable program code for loading at least one operand from the command queue to the local memory;

computer readable program code for performing at least one operation on the at least one operand to generate a result in the local memory; and

computer readable program code for storing the result generated in the local memory in the command queue.

72. (Original) A computer program product for operating a cryptographic data processing system that comprises a host processor, a system memory coupled to the host processor, and a cryptographic processor integrated circuit that is coupled to the host processor and the system memory, the computer program product comprising:

a computer readable program medium having computer readable program code embodied therein, the computer readable program code comprising:

computer readable program code for providing a command queue in the system memory;
computer readable program code for loading a command block into the command queue using the host processor;

computer readable program code for setting a value of an interrupt field in the command block to request an interrupt when the command block has been executed;

computer readable program code for executing the command block using the cryptographic processor; and

computer readable program code for invoking an interrupt using the cryptographic processor after executing the command block if the interrupt field in the command block is set to the value to request the interrupt.

73. (Original) A computer program product as recited in Claim 72, further comprising:

computer readable program code for storing error information in the command block that is associated with executing the command block using the cryptographic processor.

74 - 76. (Canceled)

77. (Original) A computer program product for operating a data processing system that comprises a host processor, a system memory coupled to the host processor, and an adjunct processor integrated circuit that is coupled to the host processor and the system memory, the computer program product comprising:

a computer readable program medium having computer readable program code embodied therein, the computer readable program code comprising:

computer readable program code for providing a command queue in the system memory;
computer readable program code for loading a command block into the command queue using the host processor, the command block comprising an input data field that contains input data;

computer readable program code for performing an operation based on the input data using the adjunct processor to generate a result; and

computer readable program code for storing the result in the input data field such that at least a portion of the input data is overwritten.

78. (Original) A computer program product as recited in Claim 77, wherein the data processing system comprises a cryptographic data processing system, the adjunct processor integrated circuit comprises a cryptographic processor integrated circuit, and the computer readable program code for performing the operation based on the input data comprises:

computer readable program code for performing a hash operation based on the input data using the cryptographic processor to generate a hash value.

79. (Original) A computer program product as recited in Claim 78, wherein the computer readable program code for storing the result in the input data field comprises:

computer readable program code for storing the hash value in the input data field such that the at least a portion of the input data is overwritten.

80. (Original) A computer program product as recited in Claim 78, wherein the command block further comprises an input pointer field that contains an address in the system memory of an incoming packet and wherein the computer readable program code for performing the hash operation comprises:

computer readable program code for performing the hash operation based on the input data and the incoming packet using the cryptographic processor to generate the hash value.

81. (Original) A computer program product as recited in Claim 80, wherein the command block further comprises an output pointer field that contains an address in the system memory for storing a decrypted packet, the computer program product further comprising:

computer readable program code for decrypting the incoming packet using the cryptographic processor to generate the decrypted packet;

computer readable program code for attaching the hash value to the decrypted packet;
and

computer readable program code for storing the decrypted packet with the attached hash value at the address in the system memory contained in the output pointer field.

82. (Original) A computer program product for operating cryptographic data processing system that comprises a host processor, a system memory coupled to the host processor, and a cryptographic processor integrated circuit that comprises a local memory and is coupled to the host processor and the system memory, the computer program product comprising:

a computer readable program medium having computer readable program code embodied therein, the computer readable program code comprising:

computer readable program code for providing a command queue in the system memory;
computer readable program code for providing a read address for the command queue
and a write address for the command queue;

computer readable program code for loading a random number sample into the command queue using the cryptographic processor beginning at the write address; and

computer readable program code for reading the random number sample using the host processor beginning at the read address.

83. (Original) A computer program product as recited in Claim 82, wherein the computer readable program code for loading the random number sample into the command queue using the cryptographic processor beginning at the write address comprises:

computer readable program code for determining if the write address plus an amount corresponding to a size of a single random number sample equals the read address; and

computer readable program code for loading the random number sample into the command queue using the cryptographic processor beginning at the write address if the write address plus the amount corresponding to the size of the single random number sample does not equal the read address.

84. (Original) A computer program product as recited in Claim 83, further comprising:

computer readable program code for incrementing the write address by the amount corresponding to the size of a single random number sample using the cryptographic processor if the write address plus the amount corresponding to the size of the single random number sample does not equal the read address, the computer readable program code for incrementing being responsive to the computer readable program code for loading the random number sample into the command queue using the cryptographic processor beginning at the write address.

85. (Original) A computer program product as recited in Claim 82, wherein the computer readable program code for reading the random number sample using the host processor beginning at the read address comprises:

computer readable program code for determining whether the read address is equal to the write address; and

computer readable program code for reading the random number sample using the host processor beginning at the read address if the read address is not equal to the write address.

86. (Original) A computer program product as recited in Claim 85, further comprising:

computer readable program code for incrementing the read address by an amount corresponding to a size of a single random number sample using the host processor, the computer readable program code for incrementing being responsive to the computer readable program code for reading the random number sample using the host processor beginning at the read address.

In re: Blaker et al.
Serial No.: 09/852,562
Page 31

APPENDIX B ~ EVIDENCE APPENDIX

None

In re: Blaker et al.

Serial No.: 09/852,562

Page 32

APPENDIX C – RELATED PROCEEDINGS APPENDIX

None.